Amendment to the Claims

1) (Currently Amended) A method of detecting congestion, comprising:

setting a minimum threshold, a maximum threshold and a feedback threshold ~~at least one threshold value~~ for buffer occupancy;

marking outgoing packets when said average queue size is between said minimum threshold and said feedback threshold;

dropping incoming packets and marking said outgoing packets when said average queue size is between said feedback threshold and said maximum threshold; and

dropping all of said incoming packets when an average queue size exceeds said maximum threshold ~~and dropping all packets when an average queue size exceeds said threshold~~.

2) (Canceled) Please cancel claim 2.

3) (Canceled) Please cancel claim 3.

4) (Canceled) Please cancel claim 4.

5) (Canceled) Please cancel claim 5.

6) (Currently Amended) The method according to claim 1 ~~5~~, wherein said average queue size is a size of a bucket of a token bucket filter.

7) (Currently Amended) The method according to claim 1 ~~5~~, wherein said packets are marked deterministically in said step of marking outgoing packets when said average queue size is between said minimum threshold and said feedback threshold.

8) (Currently Amended) The method according to claim 1 ~~5~~, wherein said packets are dropped probabilistically in said step of dropping incoming packets when said average queue size is between said feedback threshold and said maximum threshold.

9) (Currently Amended) The method according to claim 1 ~~5~~, wherein said step of marking outgoing packets when said average queue size is between said minimum threshold and said feedback threshold comprises setting at least one bit in at least one of said packets, and said step of marking said outgoing packets when said average queue size is between said feedback threshold and said maximum threshold comprises setting at least one bit in at least one of said packets.

10) (Currently Amended) The method according to claim 1 ~~5~~, further comprising the step of calculating said average queue size based on a moving average.

11) (Currently Amended) The method of according to claim 1 ~~5~~, further comprising the steps of:

varying a number of tokens consumed by a data packet; and

transmitting said packet if the number of tokens consumed by said packet is less than or equal to available tokens.

12) (Original) The method according to claim 9, wherein said bit is part of a type of services byte.

13) (Original) The method according to claim 9, wherein a core node performs said step of setting said bit.

14) (Currently Amended) The method according to claim 9, wherein said bit is a local congestion notification ~~an LCN~~ bit.

15) (Original) The method according to claim 10, wherein said moving average is an exponential moving average.

16) (Original) The method according to claim 11, wherein said step of varying the number of tokens consumed by a data packet comprises varying the number of tokens consumed by data of unit size.

17) (Original) The method according to claim 11, wherein said step of varying said number of tokens comprises the step of decreasing data of unit size monotonically if demand increases monotonically during a congestion free period.

18) (Original) The method according to claim 11, wherein said step of varying said number of tokens comprises the step of increasing data of unit size upon receipt of a message.

19) (Original) The method according to claim 12, wherein said type of services byte is a differentiated services code point byte.

20) (Original) The method according to claim 15, wherein said packets are marked deterministically in said step of marking outgoing packets when said average queue size is between said minimum threshold and said feedback threshold, wherein said packets are dropped probabilistically in said step of dropping incoming packets when said average queue size is between said feedback threshold and said maximum threshold, and wherein said step of marking outgoing packets when said average queue size is between said minimum threshold and said feedback threshold comprises a core node setting at least one bit in at least one of said packets and said step of marking outgoing packets when said average queue size is between said feedback threshold and said maximum threshold comprises setting at least one bit in at least one of said packets, wherein said bit is part of a type of services byte.

21) (Original) The method according to claim 18, wherein said step of varying said number of tokens comprises further decreasing said data of unit size if an average queue size is greater than a demand threshold.

22) (Currently Amended) A method of regulating traffic flow between nodes, comprising:

detecting congestion;

sending a message to at least one node;

regulating at least one traffic rate of said at least one node by varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, comprising the steps of:

decreasing said data of unit size monotonically if demand increases monotonically during a congestion free period;

further decreasing said data of unit size if an average queue size is greater than a demand threshold;

increasing said data of unit size upon receipt of a message; and

transmitting said packet if a number of tokens consumed by said packet is less than available tokens, wherein said at least one packet is marked deterministically; and

detecting when congestion is clear.

23) (Original) The method according to claim 22, further comprising the step of incrementing said at least one traffic rate at random times when said congestion is clear.

24) (Original) The method according to claim 22, wherein a core node detects said congestion, and

an output node sends said message to at least one input node.

25) (Original) The method of according to claim 22, wherein said step of regulating at least one traffic rate of said at least one node, comprises:

reducing said at least one traffic rate of said at least one node proportional to the amount of traffic that said at least one node is injecting when said congestion is detected.

26) (Canceled) Please cancel claim 26.

27) (Original) The method according to claim 22, wherein said step of sending a message to at least one node comprises marking at least one packet.

28) (Canceled) Please cancel claim 28.

29) (Canceled) Please cancel claim 28.

30) (Currently Amended) ~~The method according to claim 26, wherein said step of varying said number of tokens comprises~~ A method of regulating traffic flow between nodes, comprising:

detecting congestion;

sending a message to at least one node;

regulating at least one traffic rate of said at least one node, wherein said step of regulating at least one traffic rate of said at least one node, comprises:

varying a number of tokens consumed by a data packet by decreasing data of unit size monotonically if demand increases monotonically during a congestion free period; and

transmitting said packet if a number of tokens consumed by said packet is less than or equal to available tokens; and

detecting when congestion is clear.

31) (Currently Amended) ~~The method according to claim 26, wherein said step of varying said number of tokens comprises~~ A method of regulating traffic flow between nodes, comprising:

detecting congestion;

sending a message to at least one node;

regulating at least one traffic rate of said at least one node, wherein said step of regulating at least one traffic rate of said at least one node, comprises:

varying a number of tokens consumed by a data packet by increasing data of unit size upon receipt of a message; and

transmitting said packet if a number of tokens consumed by said packet is less than or equal to available tokens; and

detecting when congestion is clear.

32) (Currently Amended) ~~The method according to claim 27,~~ A method of regulating traffic flow between nodes, comprising:

detecting congestion;

sending a message to at least one node by marking at least one packet, wherein said packet is marked deterministically;

regulating at least one traffic rate of said at least one node; and

detecting when congestion is clear.

33) (Currently Amended) ~~The method according to claim 27,~~ A method of regulating traffic flow between nodes, comprising:

detecting congestion;

sending a message to at least one node by marking at least one packet, wherein said step of marking packets comprises setting at least one bit in at least one of said packets;

regulating at least one traffic rate of said at least one node; and

detecting when congestion is clear.

34) (Original) The method according to claim 30, wherein said step of varying said number of tokens comprises further decreasing said data of unit size if an average queue size is greater than a demand threshold.

35) (Original) The method according to claim 33, wherein said bit is part of a type of services byte.

36) (Currently Amended) A method of controlling traffic flow in a differentiated services domain, comprising:

determining available bandwidth by calculating an average queue size at a token bucket filter;

determining if said average queue size is greater than a demand threshold;

varying a number of tokens consumed by a data packet; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

37) (Original) The method according to claim 36, wherein said step of varying the number of tokens consumed by a data packet comprises varying the number of tokens consumed by data of unit size.

38) (Original) The method according to claim 36, wherein said number of tokens consumed by a data packet is varied based on state and demand.

39) (Canceled) Please cancel claim 39.

40) (Currently Amended) ~~The method according to claim 36,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet, wherein said step of varying said number of tokens consumed by a data packet comprises a step of decreasing data of unit size monotonically if demand increases monotonically during a congestion free period; and

transmitting said data packet if a number of tokens consumed by said data packet is less than available tokens.

41) (Currently Amended) ~~The method according to claim 36,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet, wherein said step of varying said number of tokens comprises a step of increasing data of unit size upon receipt of a message; and

transmitting said data packet if a number of tokens consumed by said data packet is less than available tokens.

42) (Original) The method according to claim 37, wherein said step of varying the number of tokens consumed by data of unit size comprises varying the number of tokens consumed by a byte of data.

43) (Original) The method according to claim 37, wherein said step of varying the number of tokens consumed by data of unit size, comprises varying the number of tokens consumed by a bit of data.

44) (Currently Amended) ~~The method according to claim 37,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, wherein a minimum number of tokens consumed by said data of unit size equals a depth of a token bucket filter divided by a min-max fair share of the buffer; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

45) (Currently Amended) ~~The method according to claim 37,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, wherein a minimum number of tokens consumed by said data of unit size equals a depth of a token generation rate divided by a min-max fair share of the bandwidth; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

46) (Currently Amended) ~~The method according to claim 37,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, wherein a maximum number of tokens consumed by said data of unit size equals a depth of a token bucket filter divided by a maximum transmission unit; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

47) (Currently Amended) ~~The method according to claim 37,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, wherein a maximum number of tokens consumed by said data of unit size equals a token generation rate divided by a minimum data buffer drain rate; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

48) (Currently Amended) ~~The method according to claim 37,~~ A method of controlling traffic flow in a differentiated services domain, comprising:

varying a number of tokens consumed by a data packet by varying the number of tokens consumed by data of unit size, wherein said step of varying the number of tokens consumed by data of unit size, comprises the steps of:

decreasing said data of unit size monotonically if demand increases monotonically during a congestion free period;

further decreasing said data of unit size if an average queue size is greater than a demand threshold; and

increasing said data of unit size upon receipt of a message, wherein said number of tokens consumed by a data packet is varied based on state and demand; and

transmitting said data packet if a number of tokens consumed by said data packet is less than or equal to available tokens.

49) (Original) The method according to claim 38, wherein said state is a congestion state.

50) (Original) The method according to claim 40, wherein said step of varying said number of tokens comprises further decreasing said data of unit size if an average queue size is greater than a demand threshold.

51-57) (Canceled) Please cancel claims 51 through 57.